

**REMARKS**

Claims 1-10 are all the claims pending in the application. Claims 1-3 and 5-9 stand presently rejected under 35 U.S.C. § 102(b) as being anticipated by Bennett et al (US Patent No. 5,189,733). Further, claim 4 is rejected under 35 U.S.C. § 103(a) as being unpatentable over Bennett in view of Finni (US Patent No. 5,941,978) and claim 10 is rejected under 35 U.S.C. § 103(a) as being unpatentable over Bennett in view of Hayball et al (US Patent No. 6,233,610). By this Amendment, Applicants amend independent claims 1, 7, and 9.

As an initial matter, Applicants respectfully request the Examiner to approve the proposed drawing corrections to Fig. 2 filed on March 15, 2002.

Amended independent claim 1 is directed to a method, which comprises, as one of the recited steps, checking whether a requested object (MO\*) is stored in a memory “of a network element connected to a digital communications network” (emphasis added).

In item 1 of the present Office Action, the Examiner refers to Bennett’s disclosure in col. 7, ln. 6-24, as teaching all the limitations recited in claim 1.

A brief summary of the disclosure of the Bennett reference can be found on page 4, third paragraph, of Applicants’ previous Amendment filed on March 15, 2002. In particular, col. 7, ln. 18-24, which refer to Fig. 9 of the Bennett reference and which were cited by the Examiner, disclose that “[w]hen an object is called ... in step 901, the routine 900 determines if the object is non-resident, i.e., not present in the main memory 102”. However, as taught in col. 4, ln. 7-16, and as shown in Fig. 1B of the reference, the main memory 102 is a memory of a computer

system 100 (preferably an IBM compatible personal computer), and not a memory “of a network element connected to a digital communications network”, as claimed in amended claim 1. In fact, as emphasized on page 5, last paragraph, to page 6, first paragraph, of Applicants’ Amendment of March 15, 2002, there is no disclosure of any network anywhere in the Bennett reference. Rather, the Bennett reference teaches a self-contained solution to limited available memory when using a software application program (e.g. word processor, spreadsheet) for a single computer 100 (col. 1, ln. 23-25).

For at least these reasons, Applicants submit that amended claim 1 is patentable over the prior art made of record.

Amended independent claim 7 is directed to a network element, which comprises, among other things, a memory, wherein the network element is connected to a digital communications network, and wherein a controller checks whether a requested object is stored in the memory (of the network element connected to the digital communications system).

Amended independent claim 9 is directed to a digital network with network elements, wherein each network element comprises, among other things, a memory, wherein each network element is connected to the digital communications network, and wherein a controller checks whether a requested object is stored in the memory (of the network element connected to the digital communications network).

Therefore, Applicants submit that arguments analogous to those presented in connection with the discussion of the patentability of amended claim 1 apply to amended claims 7 and 9

AMENDMENT UNDER 37 C.F.R. § 1.116  
US Appln. No. 09/328,893

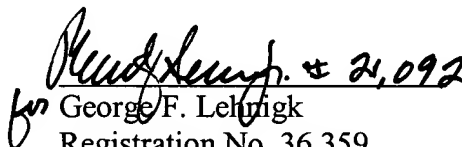
with equal force. The dependent claims should be patentable at least by virtue of their dependency from their respective independent claims.

In view of the above, reconsideration and allowance of this application are now believed to be in order, and such actions are hereby solicited. If any points remain in issue which the Examiner feels may be best resolved through a personal or telephone interview, the Examiner is kindly requested to contact the undersigned at the telephone number listed below.

The USPTO is directed and authorized to charge all required fees, except for the Issue Fee and the Publication Fee, to Deposit Account No. 19-4880. Please also credit any overpayments to said Deposit Account.

Respectfully submitted,

SUGHRUE MION, PLLC  
2100 Pennsylvania Avenue, N.W.  
Washington, D.C. 20037-3213  
Telephone: (202) 293-7060  
Facsimile: (202) 293-7860

 # 21,092  
for George F. Lehning  
Registration No. 36,359

Date: August 28, 2002



RECEIVED  
SEP 04 2002  
Technology Center 2100

APPENDIX

VERSION WITH MARKINGS TO SHOW CHANGES MADE

IN THE CLAIMS:

The claims are amended as follows:

1. (Amended) A method (100) ~~of managing a network element using managed objects (MO1, MO2, MO\*) wherein the network element is managed in response to requests (RQ) by accessing a memory (MEM) and using the objects (MO1, MO2, MO\*) stored therein, said method (100) comprising the steps of:~~

checking in response to a request (RQ = RQ\*) for access to one (MO\*) of ~~the~~ a plurality of managed objects (MO1, MO2, MO\*) whether this requested object (MO\*) is stored in the a memory (MEM) (step 110) of a network element connected to a digital communications network;

if this requested object (MO\*) is not stored in the memory (MEM), checking whether there is sufficient memory space to write this object (MO\*) into the memory (MEM) (step 120);

if there is no sufficient memory space, swapping at least one (MO1) of the stored objects (MO1, MO2) out of the memory (MEM) to a database (DB) according to at least one predeterminable criterion (step 130); and

reading the requested object (MO\*) from the database (DB) and writing it into the memory (MEM) (step 140).

wherein the network element is connected to the digital communications network,

7. (Amended) A network element for a digital communications network comprising a controller (FLT) for managing the network element using managed objects (MO1, MO2, MO\*), a memory (MEM) connected to the controller (FLT), and a database (DB) connected to the controller (FLT), wherein the network element is connected to the digital communications network, wherein the controller (FLT), in response to requests (RQ), manages the network element by accessing the memory (MEM) and using the objects (MO1, MO2, MO\*) stored therein, wherein, in response to a request (RQ = RQ\*) for access to one (MO\*) of the managed objects (MO1, MO2, MO\*), the controller (FLT) checks whether this requested object (MO\*) is stored in the memory, wherein, if this requested object (MO\*) is not stored in the memory (MEM), the controller (FLT) checks whether there is sufficient memory space to write this object (MO\*) into the memory (MEM), wherein, if there is no sufficient memory space, the controller (FLT) causes at least one (MO1) of the stored objects (MO1, MO2) to be swapped out of the memory (MEM) to a database (DB) according to at least one predeterminable criterion, and wherein the controller (FLT) reads the requested object (MO\*) from the database (DB) and writes it into the memory (MEM).

9. (Amended) A digital communications network with network elements, each network element comprising a controller (FLT) for managing the network element using managed objects (MO1, MO2, MO\*), a memory (MEM) connected to the controller (FLT), and a database (DB) connected to the controller (FLT), wherein each network element is connected to the digital communications network, wherein the controller (FLT), in response to requests (RQ), manages the network element by accessing the memory (MEM) and using the objects (MO1, MO2, MO\*) stored therein, wherein, in response to a request ( $RQ = RQ^*$ ) for access to one (MO\*) of the managed objects (MO1, MO2, MO\*), the controller (FLT) checks whether this requested object (MO\*) is stored in the memory, wherein, if this requested object (MO\*) is not stored in the memory (MEM), the controller (FLT) checks whether there is sufficient memory space to write this object (MO\*) into the memory (MEM), wherein, if there is no sufficient memory space, the controller (FLT) causes at least one (MO1) of the stored objects (MO1, MO2) to be swapped out of the memory (MEM) to a database (DB) according to at least one predeterminable criterion, and wherein the controller (FLT) reads the requested object (MO\*) from the database (DB) and writes it into the memory (MEM).